Beschreibung der Aufgabenstellung

Aufgabenstellung

Die Aufgabenstellung des hier beschriebenen Studienprojektes sieht vor, eine Online-Applikation bzw. Webapplikation zur Anforderungsaufnahme zu entwickeln. Anforderungen sollen mit Hilfe der „Rupp-Schablone“ von Chris Rupp formuliert werden. Die Idee hinter dem Online-Applikationskonzept ist die einfache Möglichkeit des zentralen Managements der Software und der guten Zugänglichkeit für mehrere Anwender auf die gleiche Datenbasis.

Als Hilfestellung zur Bewältigung dieses Projekts liegt eine Kopie des vorangegangenen Studienprojektes vor. Dieses behandelt die Aufnahme von Anforderungen auf Basis der „Rupp-Schablone“ in einer Offline-Applikation und steht zu Analyse des softwareseitigen Aufnahmeprozesses zur Verfügung.

Rupp-Schablone

Die „Rupp-Schablone“ dient der vereinfachten Aufnahme komplexer Anforderungen. Sie nutzt dazu nur eine Teilmenge der natürlichen Sprache. Genutzt wird die Schnittmenge, welche gleichermaßen Effizienz in der Bearbeitung sowie Verständlichkeit für den Anwender darstellt.

<Bild Schnittmenge>

Der Aufbau der Schablone ist an den deutschen Satzbau angepasst. Dies kann mit einer eingeschränkten Form der Subjekt-Prädikat-Objekt Logik betitelt werden. Das erste Element des Aufbaus ist ein Subjekt. Dieses Subjekt stellt den Satzgegenstand dar. Folgend wird die Wichtigkeit der auszuführenden Tätigkeit auf Basis eines helfenden Prädikats festgelegt. Dieses wird von Rupp als rechtlich bindend bezeichnet (auf Vertragsbasis). Das dritte Element ist ein optionales Dativ-Objekt zur Bestimmung des Empfängers bzw. des Nutznießers der Tätigkeit. Mit dem vierten Element wird zwischen einer selbstständigen, oder einer nutzerinitiierten Durchführung unterschieden. An fünfter Stelle steht das vom Subjekt zu behandelnde Objekt und abschließend das eigentlich auszuführende Prädikat. Das Prädikat beschreibt den Umgang bzw. die Beeinflussung des Objekts durch das Subjekt.

<Bild Schablone>

Erweiterungen

Als wünschenswerte Entwicklungen für die Webapplikation seien im Gegensatz zur derzeitigen offline Lösung des vorigen Studienprojekts genannt: ein Organisationssystem zur Zusammenarbeit mehrerer Anwender an einem Anforderungsprojekt, ein Loginsystem zur Authentifizierung der einzelnen Anwender, eine Exportfunktion für die Anforderungen zur Weiterbearbeitung offline und einige Zusatzattribute zu jeder Anforderung (Priorität, Status, Identifikationsnummer, Abhängigkeiten).

Entwicklung

Allgemein

Die Entwicklung der Webapplikation wird in mehreren Schritten durchgeführt. Vorangehend ist eine ganzheitliche Planung und Strukturierung für den erfolgreichen Abschluss des Projekts unabdinglich. Vorausschauend muss die vorgegebene Zeitspanne zur Erfüllung der Aufgabe eingeteilt und unterteilt werden. Es muss ein Konzept und ein Leitfaden entstehen, auf welchem die Entwicklung basiert und durchgeführt wird. Weiterhin müssen Überlegungen zu möglichen Technologien und deren Vor- und Nachteile getätigt werden. Sobald diese Grundbausteine ausgearbeitet sind, ist es wichtig, eine funktionierende Infrastruktur zur Softwareentwicklung zu entwerfen. Innerhalb dieser muss auch die Lauffähigkeit der zu entwickelnden Applikation gewährleistet sein. Damit kann schon jetzt später möglicherweise auftrenden Fehlern vorgebeugt, bzw. diese beschränkt werden. Nach der darauffolgenden Erstellung der Aufgaben werden diese aufgeteilt, um paralleles Arbeiten effizient zu ermöglichen.

Realisierung

Einleitung

Am Anfang der Realisierung steht die Planung sowie die Auswahl der Technologien, Programmiersprachen, Tools bzw. Tool-Chain. Diese sollten sorgsam gewählt werden, da sie die gesamte Entwicklungsdauer über genutzt werden.   
Durch die Analyse der benötigten Architektur und Infrastruktur lassen sich die verfügbaren Tools und Programmiersprachen eingrenzen, welche mit der Umgebung kompatibel sind.

Architektur

Eine Webapplikation zeichnet sich dadurch aus, dass sie im Netzwerk arbeitet. Das bedeutet, dass sie für alle authentifizierten Anwender in diesem Netzwerk zugänglich sein muss. Aus Sicht des Systems ist der Anwender ein Client, welcher Datenpakete an den Server sendet und abfragt. Basierend auf dieser Tatsache wird abgeleitet, dass es hier um eine Client-Server Architektur handelt. Der Server ist deshalb ein Webserver, der die Dokumente für den Client bereitstellt und für Interaktionen mit dem Anwender bereit steht. Da Daten persistent gespeichert werden müssen, so dass über einen längeren Zeitraum mit ihnen gearbeitet werden kann, wird eine Datenbank benötigt. Diese beiden Komponenten (Webserver, Datenbank) sind für die Datenverarbeitung zuständig, während der Client (Browser) die verarbeiteten Daten empfängt und neue Daten / Anfragen an den Server sendet.

Sprachen

Die oben beschriebene vorausgesetzte Umgebung lässt sich am einfachsten mittels XAMPP (Windows, OSX), oder LAMP (Linux) realisieren. XAMPP steht für Apache, MySQL, PHP, Perl und LAMP für Linux, Apache, MySQL, PHP. In beiden dieser Pakete sind jeweils schon ein Webserver (Apache) und eine Datenbank (MySQL, relational) enthalten. Damit die Datenbankanbindung gewährleistet ist, bieten diese Pakete PHP-Unterstützung mit an. Nur so lässt sich die Datenbank ansprechen und es können Daten ausgelesen und abgespeichert werden. Das Frontend wird dementsprechend mittels der Struktursprache HTML geschrieben. HTML erlaubt das Einbetten von (u.A.) PHP Code, falls die PHP-Unterstützung des Webservers aktiviert ist. Die Formatierung der HTML-Struktur wird durch CSS realisiert.

Zur intuitiveren Bedienung der Webapplikation wird zusätzlich das Javascript genutzt. Es erlaubt DOM-Manipulationen (Document Object Model) einfach durchzuführen.

Frameworks

Frameworks stellen schon vorgefertigte Funktionen einer bestimmten Sprache zur Verfügung. In einem größeren Softwareprojekt lässt sich so ohne Mehraufwand, im Vergleich zur puren Programmiersprache, ein funktionales Code-Grundgerüst bauen.

Das Design der Website wird mittels CSS erstellt und setzt auf ein schon strukturiertes HTML-Dokument auf. Derzeit werden allerdings immer mehr Websites über mobile Endgeräte aufgerufen, welche kleinere Displays und/oder Auflösungen als aktuelle Home Computer besitzen. Um das Design trotzdem für alle Geräte zu optimieren, eignet sich CSS nativ nur bedingt. Deshalb wird das CSS-Framework „SASS“ genutzt, welches eine ähnliche Syntax besitzt, jedoch mit weniger Code auskommt. Außerdem wird eine Klassenbibliothek für CSS-Klassen genutzt. Diese heißt „Bootstrap“ und ist in der Webentwicklung weit verbreitet. Bootstrap bietet viele responsive Klasse und Layouts, die z.B. alte nicht-responsive Tabellen-Layouts sehr einfach ersetzen können. Viele Klassen besitzen zusätzliche Effekte bei Hover (Mauszeiger über Element), oder Klickanimationen. Für einige bestimmte Effekte wird weiterhin das von jQuery stammende „jQuery UI“ genutzt, insbesondere für Klickanimationen.

Javascript erlaubt DOM-Manipulationen. Das ermöglicht eine intuitivere Bedienung der grafischen Weboberfläche. Um diese Manipulationen, wie z.B. Fades und Slides einfacher implementieren zu können, bietet das Framework jQuery eine vorgefertigte Funktionsliste, welche nur aufgerufen werden muss.

Tools